

# 064-251\_Ativ02-Cecilia-Lodron-Gonzaga.pdf

Cecilia

2025-06-14

## Exercício 1

Crie os vetores:

a.  $(1; 2; 3; \dots; 19; 20)$

R = O vetor de 1 à 20 pode ser criado de duas formas, pelo próprio comando de vetor, ou criando uma sequência:

```
c(1:20)
```

```
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

```
#comando: vetor de 1 até 20/c(início:fim)
```

```
seq(from = 1, to = 20, by = 1)
```

```
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

```
#comando: sequência de 1 até 20, aumentando 1/seq(from = primeiro, to = ultimo, by = acréscimo)
```

b.  $(20; 19; \dots; 2; 1)$

R = O vetor de 20 à 1 pode ser criado de duas formas, pelo próprio comando de vetor, ou criando uma sequência:

```
c(20:1)
```

```
## [1] 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
```

```
#comando: vetor de 20 até 1/c(início:fim)
```

```
seq(from = 20, to = 1, by = -1)
```

```
## [1] 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
```

```
#comando: sequência de 20 até 1, diminuindo 1/seq(from = primeiro, to = ultimo, by = decréscimo)
```

c.  $(1; 2; 3; \dots; 19; 20; 19; 18; \dots; 2; 1)$

R = O vetor de 1 a 20 e depois de 19 a 1 pode ser gerado criando-se um vetor que possui dois parâmetros, uma sequência de 1 à 20 e outra de 19 à 1:

```
c(1:20, 19:1)
```

```
## [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 19 18 17 16 15
## [26] 14 13 12 11 10  9  8  7  6  5  4  3  2  1
```

*#comando: vetor com a primeira variável uma sequência de 1 até 20 e a segunda variável de 19 até 1/c(c(*

d.  $(4; 6; 3)$  e atribua a ele o nome tmp.

R = Criação simples de tmp por meio do c():

```
tmp <- c(4, 6, 3)
#criando o vetor tmp
```

```
tmp
```

```
## [1] 4 6 3
```

## Exercício 2

Use a função `rep` para criar os seguintes vetores (não deixe de verificar o help dessa função):

a.  $(4; 6; 3; 4; 6; 3; \dots; 4; 6; 3)$  no qual há 10 ocorrências de 4.

R = Para criarmos o vetor acima, utilizamos o vetor `tmp` criado no exercício anterior, assim, usamos uma repetição de `tmp` dez vezes.

```
rep(tmp, 10)
```

```
## [1] 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3
```

```
#comando: repete tmp 10 vezes/rep(vetor, repetições)
```

b.  $(4; 6; 3; 4; 6; 3; \dots; 4; 6; 3; 4)$  no qual há 11 ocorrências de 4, 10 ocorrências de 6 e 10 ocorrências de 3.

R =

```
rep(tmp, length.out = 31)
```

```
## [1] 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4
```

```
#comando: repete tmp até dar 31 = 11+10+10/rep(vetor, length.out = limite)
```

c.  $(4; 4; \dots; 4; 6; 6; \dots; 6; 3; 3; \dots; 3)$  no qual há 10 ocorrências de 4, 20 ocorrências de 6 e 30 ocorrências de 3.

R =

```
rep(tmp, c(10, 20, 30))
```

```
## [1] 4 4 4 4 4 4 4 4 4 4 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 3 3 3 3 3 3 3 3
## [39] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
```

```
#comando: repete tmp de acordo com o vetor, ou seja, repete a primeira variável de tmp 10 vezes, a segun-
```

## Exercício 3

Crie um vetor dos valores de  $e^x \cos(x)$ , para  $x = 3; 3, 1; 3, 2 \dots 6$ .

R =

```
x <- seq(from = 3, to = 6, by = 0.1)
#cria uma sequência com os valores de x

c(exp(x)*cos(x))
```

```
## [1] -19.884531 -22.178753 -24.490697 -26.773182 -28.969238 -31.011186
## [7] -32.819775 -34.303360 -35.357194 -35.862834 -35.687732 -34.685042
## [13] -32.693695 -29.538816 -25.032529 -18.975233 -11.157417 -1.362099
## [19] 10.632038 25.046705 42.099201 61.996630 84.929067 111.061586
## [25] 140.525075 173.405776 209.733494 249.468441 292.486707 338.564378
## [31] 387.360340
```

```
#coloca a fórmula em um vetor, que já calcula para cada valor de x
```

## Exercício 4

Crie os seguintes vetores:

a.  $(0, 1^3x0, 2; 0, 1^6x0, 2^4; \dots; 0, 1^36x0, 2^34)$

R = Nesse exercício, podemos perceber duas sequências, as dos expoentes, e duas repetições, as dos coeficientes. Podemos colocar a fórmula de tal maneira:

```
exp1 <- seq(3, 36, by = 3)
exp2 <- seq(1, 34, by = 3)
#criamos a sequência de expoentes, que na fórmula o expoente de 0,1 é uma sequência de 3 acrescido de 3

tam <- length(exp1)
#calculando o tamanho os vetores exp1 = exp2

coef1 <- rep(0.1, tam)
coef2 <- rep(0.2, tam)
#geramos uma sequência de coeficientes baseadas no tamanho dos vetores expoentes

c((coef1**exp1)*(coef2**exp2))

## [1] 2.000000e-04 1.600000e-09 1.280000e-14 1.024000e-19 8.192000e-25
## [6] 6.553600e-30 5.242880e-35 4.194304e-40 3.355443e-45 2.684355e-50
## [11] 2.147484e-55 1.717987e-60
```

```
#vetor que calcula de forma vetorial cada valor dos expoentes e coeficientes em sequência
```

b.  $\left(2, \frac{2^2}{2}, \frac{2^3}{3}, \dots, \frac{2^{25}}{25}\right)$

R = Podemos perceber na fórmula dada que ela possui uma sequência de expoentes de 1 à 25 e uma repetição de 2 logo, semelhante ao exercício anterior, temos:

```
aux <- c(1:25)
c((2**(aux)/aux))

## [1] 2.000000e+00 2.000000e+00 2.666667e+00 4.000000e+00 6.400000e+00
## [6] 1.066667e+01 1.828571e+01 3.200000e+01 5.688889e+01 1.024000e+02
## [11] 1.861818e+02 3.413333e+02 6.301538e+02 1.170286e+03 2.184533e+03
## [16] 4.096000e+03 7.710118e+03 1.456356e+04 2.759411e+04 5.242880e+04
## [21] 9.986438e+04 1.906502e+05 3.647221e+05 6.990507e+05 1.342177e+06
```

#Exercício 5

**Calcule os seguintes somatórios:**

a.  $\sum_{i=10}^{100} (i^3 + 4i^2)$

R = Pela fórmula acima, queremos chegar em um valor real que seja o somatório da sequência de 1 até 10 da fórmula dada, logo:

```
i <- seq(from = 10, to = 100, by = 1)
sum((i**3)+(4*(i**2)))
```

```
## [1] 26852735
```

*#criamos o somatório da sequência de 10 até 100, que gera um valor real*

b.  $\sum_{i=1}^{25} \left( \frac{2i}{i} + \frac{3i}{i^2} \right)$

R = Esse exercício é semelhante ao anterior, com a diferença da fórmula e da sequência de i, que vai de 1 até 25

```
i2 <- c(1:25)
sum(((2**i2)/i2)+((3**i2)/i2**2))
```

```
## [1] 2129170437
```

## Exercício 6

Use a função `paste` para criar os seguintes vetores de caracteres de comprimento 30:

a. (“rotulo 1”, “rotulo 2”, ..., “rotulo 30”). Importante: há um único espaço entre rotulo e o número.

R = Nesse exercício, a função `paste` é utilizada para criar uma string de caracteres.

```
paste("rotulo", 1:30, sep = " ")
```

```
## [1] "rotulo 1" "rotulo 2" "rotulo 3" "rotulo 4" "rotulo 5" "rotulo 6"
## [7] "rotulo 7" "rotulo 8" "rotulo 9" "rotulo 10" "rotulo 11" "rotulo 12"
## [13] "rotulo 13" "rotulo 14" "rotulo 15" "rotulo 16" "rotulo 17" "rotulo 18"
## [19] "rotulo 19" "rotulo 20" "rotulo 21" "rotulo 22" "rotulo 23" "rotulo 24"
## [25] "rotulo 25" "rotulo 26" "rotulo 27" "rotulo 28" "rotulo 29" "rotulo 30"
```

```
#comando: paste("caracteres", repetição, sep = "separador de cada caractere")
```

b. (“fn1”, “fn2”, ..., “fn30”). Importante: não há espaço entre fn e o número

R = Exercício sememelhante ao anterior, porém com outros caracteres e sem separação

```
paste("fn", 1:30, sep = "")
```

```
## [1] "fn1" "fn2" "fn3" "fn4" "fn5" "fn6" "fn7" "fn8" "fn9" "fn10"
## [11] "fn11" "fn12" "fn13" "fn14" "fn15" "fn16" "fn17" "fn18" "fn19" "fn20"
## [21] "fn21" "fn22" "fn23" "fn24" "fn25" "fn26" "fn27" "fn28" "fn29" "fn30"
```

```
#sep = "" indica nenhum caractere para separação
```

## Exercício 7

Qual é a maneira mais conveniente para inserir um valor entre dois elementos em uma dada posição de um vetor?

R = Uma forma simples de inserir um valor em um vetor é utilizando o comando base do R `append`, na qual possui a seguinte formatação:

```
append(1:10, 11, after = 10)
```

```
## [1] 1 2 3 4 5 6 7 8 9 10 11
```

*#comando: na sequência e 1 até 10, adiciona 11 na posição posterior a posição 10*

```
append(c(1, 7, 9, 3, 4), 15, after = 3)
```

```
## [1] 1 7 9 15 3 4
```

*#por esse comando percebemos que adicionar um vetor "after = 3", o 3 é a posição e não o valor de uma po*



## Exercício 8

Seja o conjunto de dados `juul{ISwR}`

a. Quais as linhas referentes as garotas entre 7 e 14 anos de idade?

R = No conjunto de dados `juul`, do pacote `ISwR`, possui como variável idades de gatoras, na qual pode ser filtrada com o comando `subset` abaixo:

```
library(ISwR)
data(juul)
#carregando o pacote ISwR e os dados juul

u <- subset(juul, sex==2 & age>=7 & age<=14)
summary(u)
```

```
##      age      menarche      sex      igf1      tanner
## Min.   : 7.000   Min.   :1.000   Min.   :2   Min.   : 95.0   Min.   :1.000
## 1st Qu.: 8.885   1st Qu.:1.000   1st Qu.:2   1st Qu.:214.2   1st Qu.:1.000
## Median :10.560   Median :1.000   Median :2   Median :302.5   Median :1.000
## Mean   :10.557   Mean   :1.132   Mean   :2   Mean   :351.8   Mean   :1.951
## 3rd Qu.:12.207   3rd Qu.:1.000   3rd Qu.:2   3rd Qu.:473.2   3rd Qu.:3.000
## Max.   :13.940   Max.   :2.000   Max.   :2   Max.   :914.0   Max.   :5.000
##                NA's    :6                NA's    :114   NA's    :61
##      testvol
## Min.   : NA
## 1st Qu.: NA
## Median : NA
## Mean   :NaN
## 3rd Qu.: NA
## Max.   : NA
## NA's    :346
```

*#o comando summary serve para reduzir a tabela de dados da variável u, que são os dados filtrados de pe*

b. Qual a média da variável `igf1` para as garotas entre 7 e 14 anos de idade?

R = A média das variáveis filtradas no exercício anterior é:

```
mean(u$igf1, na.rm = TRUE)
```

```
## [1] 351.8233
```

*#média retirando valores NA*